CRT-2001-25

Centre de recherche sur les transports (C.R.T.)
Centre for Research on Transportation

THE SELECTIVE TRAVELING SALESMAN PROBLEM AND EXTENSIONS: AN OVERVIEW

by

Dominique Feillet
Pierre Dejax
Michel Gendreau

July 2001

102-02945/

PUBLICATION
CRT-2001-25

THE SELECTIVE TRAVELING SALESMAN PROBLEM AND EXTENSIONS: AN OVERVIEW*

by

Dominique Feillet[1]
Pierre Dejax[2]
Michel Gendreau[3]

* This research was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) and the Fonds pour la Formation de Chercheurs et l'Aide à la Recherche (FCAR).

1 Laboratoire Productique Logistique, École Centrale Paris, Grande Voie des Vignes, 92295 Chatenay-Malabry Cedex, France (feillet@pl.ecp.fr)
2 Département Automatique et Productique, École des mines de Nantes, 4, rue Alfred Kastler, BP20722, 44307 Nantes Cedex 3, France (Pierre.Dejax@emn.fr)
3 Département d'informatique et de recherche opérationnelle and Centre de recherche sur les transports, Université de Montréal, C.P. 6128, succursale Centre-ville, Montréal, Canada H3C 3J7 (michelg@crt.umontreal.ca)

Département d'informatique et de recherche opérationnelle - Publication #1202

Dépôt légal – Bibliothèque nationale du Québec
Bibliothèque nationale du Canada, 2001

ABSTRACT

The Selective Traveling Salesman Problem (STSP) is a generalization of the Traveling Salesman Problem (TSP) where it is not necessary to visit all vertices. With each vertex is associated a profit. The objective is to find a route which maximizes the collected profit in a time not exceeding a given bound. In this paper, the existing literature about the STSP is surveyed.

Key words: selective traveling salesman problem, orienteering problem, survey.

RÉSUMÉ

Le Problème du Voyageur de Commerce Sélectif (PVCS) est une généralisation du Problème du Voyageur de Commerce (PVC) pour laquelle il n'est pas nécessaire de passer par tous les sommets. A chaque nœud est associé un gain. L'objectif est de trouver une route maximisant le profit collecté en un temps limité par une borne donnée. Dans cet article, nous passons en revue la littérature existante.

Mots-clés : problème du voyageur de commerce sélectif, état de l'art.

1 Introduction

The Traveling Salesman problem (TSP) and the Vehicle Routing Problem (VRP) are among the most widely studied combinatorial optimization problems. They deal with optimally visiting customers from a central depot. Many extensions of these problems are encountered in the literature.

A special case of the TSP occurs when customers are selected according to a profit collected by visiting them. This might be seen as a bicriteria TSP. One has to maximize a profit while minimizing a travel cost. A way of considering these two objectives is to include the time objective as a constraint which limits route duration and to face the monocriterion problem of maximizing the profit. The problem is then known under the name of Selective Traveling Salesman Problem (STSP). Other monocriterion transformations of the whole problem can be found. The two objectives might be considered simultaneously in the objective function, as in the Profitable Tour Problem (e.g., Dell'Amico *et al.* 1995) and its extensions as the Prize-collecting Traveling Salesman Problem (e.g., Balas 1995). The profit objective might be seen as a constraint, compelling to collect at least as much as a given profit, leading to the Quota Traveling Salesman Problem (see Awerbuch *et al.* 1998). Feillet *et al.* (2000) propose to gather together all these kinds of problems under the name of Traveling Salesman Problems with Profits (TSPs with Profits). In this paper, we review the existing literature about the STSP. Other TSPs with Profits are surveyed in Feillet *et al.* (2000).

In section 2, we precisely define the STSP and give results about its complexity. Section 3 describes the main encountered applications. Section 4 presents heuristic solution procedures, while section 5 presents exact solution procedures. In section 6, we introduce extensions of the STSP. The extension to several vehicles is presented in section 7.

2 Definition and complexity results

Let $G = (V, A)$ be a complete graph where $V = \{v_1, \ldots, v_n\}$ is a set of n vertices and A is the set of arcs (directed STSP) or edges (undirected STSP). Let us associate a profit $p_i \geq 0$ with each vertex $v_i \in V$ (with $p_1 = 0$) and a distance $t_{ij} \geq 0$ with each arc or edge $(v_i, v_j) \in A$. In the following, we interpret vertices as customers and distances as travel times. Vertex v_1 is interpreted as a depot. The STSP consists in determining a simple circuit including vertex v_1 of maximal total profit and whose duration does not exceed a preset value t_{max}.

The travel time matrix is assumed to satisfy the triangle inequality. When the travel time matrix is symmetrical we fall into the undirected case. In this case, we will then consider the STSP on a graph $G = (V, E)$, where the set of edges E is $\{(v_i, v_j), v_i \in V, v_j \in V, i < j\}$.

The STSP is closely related to the Orienteering Problem (OP) (e.g., Tsiligirides 1984). In the OP, a destination is introduced and one looks for an optimal path from the depot to the destination. Thus, the STSP is a special case of the OP where the depot and the destination are merged. In the following, we will not separate solution procedures dedicated to the OP, as this difference does not appear to deeply affect them.

A variety of problems either merely duplicate the STSP or include it as a special case, as the Time Constrained Traveling Salesman Problem (1981), the Maximum Collection

Problem (1988), the Resource Constrained Traveling Salesman Problem (1990a) or the Traveling Salesman's Subtour Problem with One Additional Constraint (TSSP+1) (1992).

As expected, the STSP falls within the class of NP-hard problems. Golden *et al.* (1987) show this through the recognition version of the TSP: given a collection of vertices, is there a tour of length less than t_{max} through all the vertices? They consider any instance of the recognition version of the TSP, assign a profit of 1 to every vertex, set the time limit to t_{max} and choose a vertex to be the depot. Then they claim that solving the STSP for this instance will answer the question. Indeed, if the total score is equal to the number of vertices, the answer is yes, otherwise it is no.

Laporte and Martello (1990) propose another proof based on the transformation of the recognition version of the STSP from the Hamiltonian Circuit Problem, which is known to be NP-complete. Given any instance of the Hamiltonian Circuit Problem, relative to a graph $G^h = (V, A^h)$, they define an instance of the STSP assigning a profit of 1 to every vertex, a duration of 1 to every arc in A^h, a duration of 2 to every arc in $A \setminus A^h$ (where A is the complete set of arcs) and fixing a time limit equals to $|V|$. Then G^h possesses a Hamiltonian circuit if and only if the solution value of this STSP instance equals $|V|$.

3 Applications

Traveling salesman problems with selection of vertices are first introduced by Cloonan (1966) in the context of a sales territory problem. Gensch (1978) faces them for scheduling the visits of customers by specialists of a steel firm, while Davis *et al.* (1982) are concerned with scheduling branch messengers at a commercial bank.

The first appearance of the STSP as such is in Golden *et al.* (1984) where they propose a solution procedure for a vehicle routing problem with an inventory component. A fleet of trucks must deliver fuel to a large number of customers on a daily basis. Each customer has a known tank capacity and its fuel level has to remain above a critical value. A first step of the solution procedure is to determine which customers to serve each day. A forecasted tank level for each customer results in a measure of emergency. Then, finding the set of selected customers amounts to solving a STSP.

The naming of the OP is introduced by Tsiligirides (1984) in the context of orienteering. The score orienteering event is a competition that takes place in mountainous or forest areas. Competitors start from a control point and have to reach another control point within a prescribed time limit. Meantime they can visit other control points and collect scores allocated to them. The winner is the competitor ending with the maximum score.

In a chemical production context, Pekny *et al.* (1990b) try to determine schedules for a number of jobs that are candidates for processing over a given time horizon through a system where costs depend only on the consecutive jobs in the production sequence. Such systems could be flowshops with the zero wait processing condition or single machine scheduling with setup times for example. A benefit is associated to each completed job. When the objective is to maximize benefits over the horizon, it leads up to the STSP. With other objectives we fall within other TSPs with Profits.

Ramesh and Brown (1991) propose an application in control theory. Let a target location be distributed probabilistically over N cells. A searcher is available to search the cells and wants to maximize its probability to find it covering a limited distance. More precisely, the

searcher detects the target in a cell (if present) with a given probability. It can visit a given cell several times but the probability decreases at every visit. This problem can be modeled as the STSP, but no computational results are presented.

Another original application of the STSP is proposed by Göthe-Lundgren *et al.* (1996). They look at a vehicle routing cost allocation problem with a game theory approach. The computation of the nucleolus is related to solving the STSP.

Awerbuch *et al.* (1998) call the STSP without depot the bank robber problem: a robber wants to rob a maximum of money in banks, knowing the location and the amount of money in each bank. He drives a car that he cannot refuel after the first robbery, since he would be reported to the police.

We can find other applications for the STSP in the case of multi-vehicle extensions. Examples are recruiting of athletes, fisheries surveillance, aerial reconnaissance or ships replenishment. Some of these applications are detailed in section 7.

4 Heuristic solution procedures

Heuristic solution procedures developed for solving the STSP are usually composite procedures combining many steps, often extended from classical TSP procedures. The procedures presented in sections 4.2 and 4.3 are usually proposed as steps of composite procedures. They are fast but not completely effective. Composite procedures are presented further in section 4.4.

4.1 Preprocessing

The definition of the STSP implies that some vertices might not be visited. In fact, one can imagine that some vertices cannot directly be visited from the depot without exceeding the time constraint. A first step of some procedures is to identify and take out vertices such that $t_{1i} + t_{i1} > t_{max}$. Keller (1989) or Chao *et al.* (1996a) solution procedures, for example, include this first step. This preprocessing is of course also relevant in exact solution approaches.

4.2 Tour construction procedures

These procedures build up a route step by step in a greedy way. Actually, some of them either use stochastic criteria or criteria dependent on parameter values, and are repeated a number of times so as to generate as good solutions as possible.

Most of the criteria used for the addition of a vertex rely on the one hand on some intrinsic evaluations (profit,...) and on the other hand on its position regarding the current partial route. Only the notion of center of gravity measure is based on a knowledge given by a previously computed feasible route. We report three kinds of construction procedures which are described below.

4.2.1 Path extension procedures

These construction procedures consist in extending a path until no more vertices can be added, due to the time constraint. Initially the path is restricted to the depot. At each

iteration a vertex $v_i \in V$ is selected, depending on a given criterion, among the set of candidate vertices (vertices not currently on the path that can be inserted between the last vertex v_s of the path and the depot). These construction procedures are closely related to the Nearest Neighbor construction procedure proposed by Rosenkrantz *et al.* (1977) for the TSP.

Researchers propose several selection criteria:

1. Ramesh and Brown (1991) propose to select the candidate vertex $v_i \in V$ that maximizes p_i.
2. Keller (1989) and Laporte and Martello (1990) propose to select the candidate vertex $v_i \in V$ that maximizes p_i/t_{si}.
3. Tsiligirides (1984) considers the 5 candidate vertices with the highest ratio p_i/t_{si}, and then standardizes and stacks the ratio $(p_i/t_{si})^4$ of these vertices so that their sum equals one, and selects one of them in a probabilistic fashion.
4. Keller (1989) proposes to standardize and stack the profits of all candidate vertices so that their sum equals unity, and to select one of them in a probabilistic fashion.

Laporte and Martello (1990) extend this principle by allowing the construction on both ends of the path. For every of these criteria, the time complexity of the procedure is as least as good as that of the Nearest Neighbor construction procedure, i.e., $O(n^2)$. Intuitively, it appears clearly that the ratio p_i/t_{si} captures more significantly the trade-off between profit and time consumption than the myopic focus on the profit. Actually, criteria 1 and 4 are only proposed as a part of a more complex procedure.

4.2.2 Route insertion procedures

These construction procedures consist in inserting vertices in a route until no more vertices can be added. Initially the route is restricted to a loop around the depot. At each iteration a vertex $v_i \in V$ and a position in which to insert it are selected, depending on a given criterion, among the set of candidate vertices (vertices not currently on the path that can be inserted in the route, while the time constraint remains satisfied). These construction procedures are closely related to the Cheapest Insertion construction procedure proposed by Rosenkrantz *et al.* (1977) for the TSP.

Several criteria are proposed by researchers:

1. Laporte and Martello (1990) and Ramesh and Brown (1991) propose to choose the candidate vertex $v_i \in V$ that maximizes $p_i/(t_{ri} + t_{is} - t_{rs})$ and insert it between v_r and v_s, where v_r and v_s are two consecutive vertices of the route.
2. Golden *et al.* (1987) select a candidate vertex according to an order depending on:
   (a) its profit,
   (b) a distance to center-of-gravity measure (at the first iteration the center of gravity is defined as the center of gravity of all the vertices weighted by their profits, then it is set as the center of gravity of the vertices of the current route),

(c) its distance to the depot (in the case of the OP, this distance is the sum of the distances to the origin and the destination).

The selected vertex is inserted as cheaply as possible. Note that measure (2b) is the only to be recomputed after each insertion.

3. Golden *et al.* (1988) compute a measure determined for each vertex v_i by:

   (a) a score measure, given by $\sum_{v_j \in V} p_j e^{-\mu t_{ij}}$, taking into account the density of profit around the vertex v_i, where μ is a parameter chosen to reflect the scale of intervertex distances. Note that in Golden *et al.*'s (1988) composite procedure, this measure is in fact also adjusted by a learning measure, evaluating the amount of presence of the vertex v_i in routes generated during the overall algorithm (see section 4.4.1).

   (b) a distance to center-of-gravity measure (center of gravity is either the center of gravity of a previously known route, weighted by the profits, or a chosen location),

   (c) measure (2c) described above.

   A vertex is selected in a probabilistic fashion, with a uniform probability on the five candidate vertices with the largest measure. The selected vertex is inserted as cheaply as possible. Note that the measures are computed only one time for all insertions, as insertions do not affect them.

Ramesh and Brown (1991) develop the idea of a double insertion. Two candidate vertices v_i and v_j are inserted consecutively between two consecutive vertices v_r and v_s of the current route such that $(p_i + p_j)/(c_{ri} + c_{ij} + c_{js} - c_{rs})$ is maximized. This scheme manages better than the single insertion method which does not consider how the insertion affects future construction steps, but the price to pay is an increased computing time.

Gendreau *et al.* (1998b) propose the insertion of clusters. A dispersion index and a proximity measure defined on sets of vertices lead to the selection of several clusters. A cluster is evaluated as a vertex located at its center of gravity and having a profit equals to the sum of the profits of its vertices. A specialized insertion heuristic for the TSP is applied to insert the selected cluster.

Although their processing times exceed path extension's, route insertion procedures are still fast and reach better solution values, if we are to believe their predominant use in effective heuristics. Indeed, path extension amounts to take one step forward along a route with a myopic view focused on the next destination, while route insertion considers the route as a whole.

4.2.3 Sweep approach procedure

Tsiligirides (1984) proposes a solution approach based upon the Sweep algorithm due to Wren and Holliday (1972) for the TSP. The geographic area is divided into sectors determined by two concentric circles and an arc of given length. Sectors are changed by varying the two radii of the circles and by rotating the arcs. Routes are built up within each sectors. 48 cases are examined and the best route is selected. This heuristic is compared with the

stochastic path extension procedure described in section 4.2.1 (criterion 3). Computing times are similar but results are clearly inferior for the sweep approach.

4.3 Tour improvement procedures

These procedures start with a feasible route and manage to improve it. We report three kinds of improvement methods, which are detailed below. Actually, the definition of what is exactly an improvement procedure is not clear, as, unlike TSP, even a null solution is feasible and all kinds of construction procedures could also be included in this family. Thus, the procedures that we describe here could be stated as procedures which involve local moves between good solutions.

4.3.1 TSP improvement procedures

A first kind of improvement is given by blindly considering the set of vertices visited by the route and trying to shorten the length of the route through these vertices. In this case, we are exactly in the situation of an improvement procedure for the classical TSP. However, the goal is not to shorten the duration of the route, which does not appear in the objective function, but to allow the future addition of some new vertices.

The TSP solution procedures used by the researchers for the STSP are the 2-opt procedure of Lin (1965) (e.g., Chao *et al.* 1996a), the 3-opt procedure of Lin (1965) (e.g., Ramesh and Brown 1991) and the GENIUS algorithm of Gendreau *et al.* (1992) (see Gendreau *et al.* 1998b). Actually, the field of improvement procedures for the TSP has seen a lot of attention and many effective procedures, mostly based on Lin's λ-opt mechanism, are available, the question being how much time one is ready to spend for improvement. However, Tsiligirides (1984) proposes his own procedure based on interchanges between pair of vertices and Keller (1989) develops his own two routines. The first one searches for and eliminates self-crossing paths, which by the triangle inequality cannot be contained in an optimal route. The second one sequentially shifts every vertex of the route to find its best position. No comparison with existing improvement procedures justifies the use of these new procedures.

4.3.2 Substitution based improvement procedures

A second kind of improvement is given by the idea of substitution. A main feature of the STSP is that a feasible solution is restricted to visit a subset of vertices. Thus, it might be improved by replacing some visited vertices with more profitable ones. Tsiligirides (1984) proposes to insert a vertex in the route while another one, with an inferior profit, is suppressed, provided that the route keeps feasible. Keller (1989) extends this scheme by authorizing the deletion of two consecutive vertices.

Ramesh and Brown (1991) apply a different method, which searches for the best vertex to delete. It assesses the deletion of every vertex followed by some 2-opt interchanges. The vertex which minimizes the profit-to-savings ratio is deleted. It remains to insert new vertices as effectively as possible. This procedure is not exactly an improvement procedure as it does not ensure an improvement of the objective function, but rather searches for

moves in promising directions. A danger of this method is that cyclical generation of routes might occur, which can be prevented by keeping track of routes generated so far.

An interesting possibility proposed by Ramesh and Brown (1991) is to allow to slightly exceed the time limit t_{max} while performing these substitutions. It is all the more effective given that feasibility is always easy to recover by suppressing vertices, or, at best, by reordering the vertices.

4.3.3 Vertex partition based improvement procedures

Unlike previous methods, Chao *et al.* (1996a) do not focus on a single route but try to improve the best route among a set of feasible routes. Vertices are partitioned in a set of feasible routes, where the route with the highest profit is emphasized. Two procedures are described, with two kinds of possible movements. In the so-called two-point exchange procedure each vertex of the best route is considered in sequence. It is possible to move it to another route while in return a vertex of this route moves to the best route. In the so-called one-point movement procedure every vertex of every route is considered in sequence. This time, it is just possible to move it to another route.

In both procedures and for each considered vertex, moves are made as soon as the highest profit among the set of routes increases. An important point is that the emphasized route might swap during the process. Another important point occurs when no move improves the solution for a given vertex. Indeed, authors allow to slightly deteriorate the solution. The best move is applied if it does not induce a deterioration larger than a given percentage of the current highest route profit. This method is named the record-to-record principle and is introduced by Dueck (1993). It may be seen as a deterministic annealing procedure. The performance of these procedures will be described later in section 4.4.

4.4 Composite procedures

In fact, most of the solution procedures found in the literature are composite procedures based on the materials described above. Three heuristic composite procedures (Golden *et al.* 1988, Ramesh and Brown 1991 and Chao *et al.* 1996a) outperform other attempts. These three heuristic solution procedures are described below. All of them are evaluated on Tsiligirides' instances (1984), involving up to 33 vertices. They appear to perform quite well both with respect to accuracy and running time, and to be nearly always optimal. Instances are typically solved in a few seconds. A detailed comparison of many procedure performances on Tsiligirides' instances is given in Chao *et al.* (1996a). Table 1 summarizes classical heuristic procedures devised for solving the STSP.

4.4.1 Golden *et al.* heuristic procedure

Golden *et al.* (1988) propose a heuristic procedure combining a previous heuristic procedure by Golden *et al.* (1987) and the Tsiligirides' randomization concept (1984).

A first route is computed with a stochastic insertion procedure (see 4.2.2). The insertion procedure stops when the time limit is exceeded. The vertex with the lowest score-to-savings ratio with a savings sufficient to reach feasibility, is then dropped. The generated route is

Reference	Name	Components
Tsiligirides (1984)	Stochastic algorithm	Path extension + improvement
Tsiligirides (1984)	Deterministic algorithm	Sweep algorithm + improvement
Golden *et al.* (1987)	Center of gravity heuristic	Route insertion, improvement, ...
Golden *et al.* (1988)	Multifaceted heuristic	See section 4.4.1
Keller (1989)	MVP algorithm	Path extension, improvement, ...
Laporte and Martello (1990)	H1	Path extension
Laporte and Martello (1990)	H2	Route insertion
Ramesh and Brown (1991)	Four-phase algorithm	See section 4.4.2
Chao *et al.* (1996a)		See section 4.4.3
Fischetti *et al.* (1998)		Adaptation of four-phase algorithm
Gendreau *et al.* (1998a)	H1, H2	Adaptations of four-phase algorithm
Gendreau *et al.* (1998b)	Insert and Shake	Route insertion + improvement

Table 1: Classical heuristic procedures for the STSP.

passed to a 2-opt improvement routine. Following this, new insertions are attempted using the insertion procedure. The outcome is a candidate solution route.

This route is used to update the center of gravity coefficient (see 4.2.2) and the learning measure. The learning measure of a vertex $v_i \in V$ consists in the average route score for routes that include v_i against the total average route score.

Five starting center-of-gravity locations are used and, for each of them, the procedure is repeated 20 times. The procedure recommends the solution route with the highest profit, among the 100 generated solution routes.

4.4.2 Ramesh and Brown four-phase heuristic procedure

Ramesh and Brown (1991) propose a heuristic procedure which consists in four phases:

1. Insertion phase. A route is built up by a series of double insertions, followed by single insertions until no more insertion is possible (see 4.2.2). The time constraint might be violated as the limit considered for the insertions is $\lambda_1 t_{max}$ with $\lambda_1 \geq 1$ instead of t_{max}.

2. Improvement phase. The 2-opt improvement procedure is carried out for each pair of edges in the tour. If the improvement has gone beyond a given threshold λ_2, the 3-opt procedure is also carried out. If the total improvement has gone beyond another threshold λ_3, the algorithm returns to phase 1 (with a reduction of λ_1 when $\lambda_1 > 1$).

3. Deletion phase. A vertex is deleted (see 4.3.2). The algorithm comes back to phase 1.

4. Final insertion phase. A stopping criterion based on the improvement between two successive iterations and the total number of iterations is included in the previous phases. When this criterion is met, insertions are performed until the time limit is reached (see 4.2.1).

Fischetti *et al.* (1998) adapt and simplify this heuristic to use it inside a Branch and Cut procedure. They also integrate as a first step an insertion system based on the value

of arcs in their current relaxed solution. Gendreau *et al.* (1998a) also propose two related heuristics inside a Branch and Cut procedure. These heuristics perform efficiently and effectively but do not attempt to introduce any major new concept.

4.4.3 Chao *et al.* heuristic

Chao *et al.* (1996a) develop a solution procedure based on the record-to-record principle.

At the beginning, the unreachable vertices due to the time limit are suppressed. In a first step, routes are constructed with no regard to the score of vertices. A first route is built up by inserting vertices in a cheapest fashion regarding the distance. When the time limit prevents from inserting new vertices, another route is built up, and so on until every vertex is covered. To diversify the approach, the process is repeated 10 times, the first route beginning each time with one of the 10 vertices farthest from the depot. The procedure keeps the set of routes which contains the route with the highest score.

An improvement procedure using the record-to-record principle is then applied to this set of routes (see 4.3). During this phase, the algorithm allows moves which deteriorate the solution no more than 10% and the 2-opt procedure is applied. This phase is repeated 10 times.

At the end, a set of vertices (one the first time, two the second time,...) is deleted from the optimal route, inserted into other routes, and the improvement step is applied again. Deleted vertices are vertices with the lowest profit-to-savings ratio. The whole process is repeated 10 times. The procedure returns the generated solution with the highest profit.

4.5 Metaheuristic procedures

Metaheuristic procedures are the core of several recent investigations. As far as we are aware, three main types of metaheuristics have been applied to the STSP. A tabu search and a neural network approaches are described below. Besides them, Smith and Tasgetiren (1999) propose a genetic algorithm with a random key representation. Moreover, composite procedures by Ramesh and Brown (1991) (see section 4.4.2) and Chao *et al.* (1996a) (see section 4.4.3) may be interpreted as metaheuristic procedures, since they allow deteriorating or infeasible intermediary solutions during the search process. The same goes for the composite procedure by Golden *et al.* (1988) in which the learning measure concept anticipates ant algorithms. This section concludes with table 2, which shortly recapitulates metaheuristic approaches for solving the STSP.

4.5.1 Tabu search

Gendreau *et al.* (1998b) propose a tabu search procedure. Moves consist either in the deletion of a portion of route, the insertion of a vertex or the insertion of a cluster according to the value of a profit-to-savings ratio. The authors introduce a parameter α reflecting the relative importance attributed to length and profit of a route so as to balance between insertion and deletion. Thus, deletion is favoured by a low ratio while insertion is favoured by a high one. At every iteration, α is modified to favour insertion or deletion whether the route is feasible or not. This local search scheme is embedded in a tabu search procedure in order to avoid cycling. When a vertex is removed from the route, it is assigned a tabu status

for a randomly selected number of iterations. Furthermore, an insertion procedure and the improvement algorithm GENIUS (see 4.2.2) are regularly called for within the procedure.

The tabu search procedure is tested on randomly generated instances involving up to 300 vertices and always yields optimal or near optimal solutions (with a gap typically less than 1%). Computation times never go beyond a couple of minutes on a Sun Sparcstation 1000. Even if no results indicate direct comparisons with other heuristic procedures, tabu search seems to emerge among the most effective and efficient approaches.

4.5.2 Neural network approach

Wang *et al.* (1995) apply a continuous Hopfield neural network method for solving the STSP. The neural network is conceptualized as a matrix with each row corresponding to a vertex and each column corresponding to a position in the path. A last column is added to determinate selected vertices and calculate the total score of the path.

The energy function is a fourth order convex function. It consists of six terms that penalize columns with more than one activated vertex, solutions with fewer than n vertices and solutions exceeding the time limit, that encourage to start and end at the depot, that match up the last column with the first n ones and that try to maximize the total score. Even if the model supports solutions with exactly n vertices, it will work efficiently for solutions with fewer than n visited vertices since solutions with consecutive visits of a same vertex are authorized with no penalty.

The links weights are not constant, each weight being the second partial derivative of the energy function with respect to the state. The model also includes a route insertion procedure and a 2-opt improvement routine that are essential factors in the success of the method.

This approach proves to be comparable to composite heuristic solution procedures of section 4.4.

Technique	Reference
Tabu search	Gendreau *et al.* (1998b)
Neural networks	Wang *et al.* (1995)
Genetic algorithm	Smith and Tasgetiren (1999)
Undefined	Golden *et al.* (1988), Ramesh and Brown (1991), Chao *et al.* (1996a)

Table 2: Metaheuristic procedures for the STSP.

5 Exact algorithms

Exact solution procedures proposed for the STSP rely on enumeration trees, efficiently managed with bound evaluations. These bounds are connected with integer linear programming formulations. In this section, we first examine these formulations, then we describe bounding schemes and related algorithms.

5.1 Integer linear programming formulations

As the STSP is closely related to the TSP, formulations mostly follow from integer linear programming formulations for the TSP, both in directed and undirected cases. Let us begin with the directed case. We slightly blend formulations to propose the following formulation, inspired by the formulation of Dantzig *et al.* (1954) for the TSP. It associates one binary variable x_{ij} to every arc $(v_i, v_j) \in A$, equal to 1 if and only if the corresponding arc is used in the solution, and one binary variable y_i to every vertex $v_i \in V$, equal to 1 if and only if the corresponding vertex is visited. The formulation is

$$\text{maximize} \sum_{v_i \in V} p_i y_i \tag{1}$$

subject to

$$\sum_{v_j \in V \setminus \{v_i\}} x_{ij} = y_i \qquad (v_i \in V), \tag{2}$$

$$\sum_{v_i \in V \setminus \{v_j\}} x_{ij} = y_j \qquad (v_j \in V), \tag{3}$$

$$\sum_{v_i \in S} \sum_{v_j \in S \setminus \{v_i\}} x_{ij} \leq |S| - 1 \qquad (S \subset V \setminus \{v_1\}, 2 \leq |S| \leq n-2), \tag{4}$$

$$\sum_{v_i \in V} \sum_{v_j \in V \setminus \{v_i\}} t_{ij} x_{ij} \leq t_{max}, \tag{5}$$

$$y_1 = 1, \tag{6}$$

$$x_{ij} \in \{0,1\} \qquad ((v_i, v_j) \in A), \tag{7}$$

$$y_i \in \{0,1\} \qquad (v_i \in V). \tag{8}$$

Constraints (2) and (3) are the so-called assignment constraints. Constraints (4) eliminate subtours that do not involve the depot. Constraint (5) enforces the time restriction.

One can find several modifications on variable definitions leading to similar formulations. Kataoka *et al.* (1998) propose the same model framework with the opposite meaning for y_i variables. Laporte and Martello (1990) let y_i variables be implicit. Packing constraints replace assignment constraints (2) and constraints (9) are added

$$\sum_{v_i \in V} x_{ik} = \sum_{v_j \in V} x_{kj} \qquad (v_k \in V) \tag{9}$$

while one might suppress constraints (3) which turn out to be redundant. Pekny *et al.* (1990b) propose to consider non-visited vertices as separate subtours of length 1 (self-loops) by introducing variables x_{ii} instead of variables y_i. This approach is convenient to emphasize the assignment problem substructure. Let us point out that all these approaches have been previously introduced in different kinds of TSPs with Profits (e.g., Gensch 1978 or Fischetti and Toth 1988). Fischetti and Toth (1988) also propose to work in a augmented graph. A dummy vertex v_0 is introduced, to which non visited vertices are connected (arc variables x_{0i} replace variables y_i). This approach points out the underlying arborescence problem substructure of TSPs with Profits. Ramesh *et al.* (1992) propose it for the undirected STSP.

Besides variable definitions, several formulations are proposed for the subtour elimination constraints. Laporte and Martello (1990) propose a formulation that we can rewrite by introducing y_i variables,

$$2\sum_{v_k \in S} y_k \leq |S|\Big(\sum_{v_i \in S, v_j \in V \setminus S} x_{ij} + \sum_{v_i \in V \setminus S, v_j \in S} x_{ij}\Big) \qquad (S \subset V \setminus \{v_1\}, |S| \geq 2) \tag{10}$$

Fischetti and Toth (1988) also propose another formulation in the context of other TSPs with Profits but valid for the STSP,

$$\sum_{v_i \in S} \sum_{v_j \in V \setminus S} x_{ij} \geq y_k \qquad (S \subset V, v_1 \in S, v_k \in V \setminus S) \tag{11}$$

Millar and Kiragu (1997) propose another kind of formulation of the STSP which combines permutation variables with flow variables. This formulation involves $2n^2$ variables and fewer than n^3 constraints. Variables x_{ij} still indicate the presence of arcs in the route, but binary variables y_{it} are introduced. $y_{it} = 1$ means vertex v_i is visited in position t on the route. The proposed formulation is

$$\text{maximize} \sum_{v_i \in V \setminus \{v_1\}} \sum_{2 \leq t \leq n} p_i y_{it} \tag{12}$$

subject to

$$\sum_{i \in V \setminus \{v_1\}} y_{it} \leq 1 \qquad (2 \leq t \leq n), \tag{13}$$

$$\sum_{2 \leq t \leq n} y_{it} \leq 1 \qquad (v_i \in V \setminus \{v_1\}), \tag{14}$$

$$\sum_{v_i \in V \setminus \{v_1\}} y_{it+1} - \sum_{v_i \in V \setminus \{v_1\}} y_{it} \leq 0 \qquad (2 \leq t \leq n-1), \tag{15}$$

$$y_{j2} - x_{1j} = 0 \qquad (v_j \in V \setminus \{v_1\}), \tag{16}$$

$$y_{it} + y_{jt+1} - x_{ij} \leq 1 \qquad ((v_i, v_j) \in V \setminus \{v_1\}, 2 \leq t \leq n-1) \tag{17}$$

$$\sum_{v_i \in V} x_{1i} - \sum_{v_i \in V} x_{i1} = 0, \tag{18}$$

$$\sum_{v_j \in V} x_{ji} - \sum_{2 \leq t \leq n} y_{it} = 0 \qquad (v_i \in V \setminus \{v_1\}), \tag{19}$$

$$\sum_{v_j \in V} x_{ij} - \sum_{2 \leq t \leq n} y_{it} = 0 \qquad (v_i \in V \setminus \{v_1\}), \tag{20}$$

$$\sum_{v_i \in V} \sum_{v_j \in V} c_{ij} x_{ij} \leq t_{max}, \tag{21}$$

$$y_{it} \in \{0,1\} \qquad (v_i \in V, 2 \leq t \leq n), \tag{22}$$

$$x_{ij} \in \{0,1\} \qquad (v_i \in V, v_j \in V). \tag{23}$$

Constraints (13) assign at most one vertex to every position on the route. Constraints (14) assign at most one position to every vertex on the route. Constraints (15) assign positions in a successive manner. Constraints (17), (19) and (20) match up permutation and flow

variables. Constraints (16) and (18) are specific to the depot. Constraint (21) is for the time restriction.

Let us now consider the undirected case. For every subset of vertices S, let $\delta(S)$ be the set of edges with one end in S and the other end in $V \setminus S$. The formulation (1)-(8) becomes, with straightforward notations,

$$\text{maximize} \sum_{v_i \in V} p_i y_i \tag{24}$$

subject to

$$\sum_{e \in \delta(\{v_i\})} x_e = 2y_i \qquad (v_i \in V), \tag{25}$$

$$\sum_{e \in S} x_e \leq |S| - 1 \qquad (S \subset V \setminus \{v_1\}, 3 \leq |S| \leq n-2), \tag{26}$$

$$\sum_{e \in E} t_e x_e \leq t_{max}, \tag{27}$$

$$y_1 = 1, \tag{28}$$

$$x_e \in \{0,1\} \qquad (e \in E), \tag{29}$$

$$y_i \in \{0,1\} \qquad (v_i \in V). \tag{30}$$

Such a formulation is found in Leifer and Rosenwein (1994). In this formulation, cycles of length 2 are not allowed. Fischetti *et al.* (1998) claim that one can assume, without loss of generality, that the optimal cycle contains at least 3 edges, as cycles of length 2 can easily be found by enumeration. Laporte and Martello (1990) propose to widen domain of variables x_e to $\{0,1,2\}$ for every $e \in E$ connected to the depot, so as to catch cycles of length 2 in the model.

Gendreau *et al.* (1998a) and Fischetti *et al.* (1998) propose to change the subtour elimination constraints (27) to

$$\sum_{e \in \delta(\{S\})} x_e \geq 2y_i \qquad (S \subset V, v_1 \in S, v_i \in V \setminus S) \tag{31}$$

As shown by the different previous formulations, one can emphasize some special features of the problem that reveal its similarities to well-known problems as the Knapsack Problem, the Assignment Problem or the Minimum 1-Tree, for which efficient solution procedures exist. Thus, by the way of relaxations, it is possible to propose bounding schemes and exact solution procedures. The three next subsections describe attempts made to develop exact solution procedures in this fashion.

5.2 The knapsack upper bound and related algorithms

Besides its similarity to the TSP, one of the main characteristics of the STSP is its Knapsack underlying substructure. This characteristic is used by Laporte and Martello (1990) in a bounding scheme, which they embed in a Branch and Bound procedure. Millar and Kiragu (1997) adapt it inside a completely different Branch and Bound procedure. These

two approaches are evaluated both on directed and undirected instances and, in both approaches, directed STSP instances were easier to solve than undirected STSP instances. This observation is not particular to the STSP, but agrees with results from several related routing problems. The bounding scheme and the two procedures are described below.

5.2.1 Laporte and Martello's Branch and Bound procedure

Laporte and Martello (1990) propose an upper bound evaluated with the solution of a knapsack problem. This bound is valid in both directed and undirected cases. To compute it, weights w_j are defined for every vertex $v_j \in V$. In the directed case the definition is

$$w_j = \alpha \min_{i \neq j} c_{ij} + (1 - \alpha) \min_{k \neq j} c_{jk}$$

with $0 \leq \alpha \leq 1$ (the definition is easy to adapt for the undirected case). With this definition, the length of a route is an upper bound for the sum of the weights of its vertices. Thus, the optimal solution of the 0-1 knapsack problem applied to vertices $v_j \in V \setminus \{v_1\}$ for maximizing the collected profit with a total weight limited to $t_{max} - w_1$, is an upper bound for the STSP.

Authors tighten this bound using the fact that vertex v_1 necessarily belongs to the routes. For every pair of vertices $v_r, v_s \in V \setminus \{v_1\}$ (with $r \neq s$), they introduce weights $w_j(r,s)$ taking account of the presence of a sequence $v_r \rightarrow v_1 \rightarrow v_s$ imposed in the route. Then, if $z(r,s)$ is the upper bound evaluated when $v_r \rightarrow v_1 \rightarrow v_s$ is in the route, $\max_{2 \leq r,s \leq n} z(r,s)$ is an upper bound for the STSP.

This upper bound is used within a Branch and Bound procedure. This procedure consists in gradually extending a path emanating from vertex v_1 through a breath-first enumeration. At a general node of the search tree, descendant nodes are generated for each vertex that can extend the current path. Another search approach proved to be inefficient. At a general node, a vertex v_j was selected and three descendant nodes were generated. In the first one, the path was extended with v_j. In the second one, v_j was enforced to be included in the path later. In the third one, v_j was definitely forbidden. An initial lower bound is given by a heuristic algorithm (see 4.2.2). This lower bound and the previous upper bound are used for fathoming nodes of the search tree. Some randomly generated instances involving between 10 and 90 vertices are solved to optimality in less than 100 seconds on a VAX 11/780 computer.

5.2.2 Millar and Kiragu's Branch and Bound procedure

Millar and Kiragu (1997) apply a Branch and Bound procedure based on the integer programming formulation they propose (see 5.1), using a standard mixed-integer solver. However, Laporte and Martello's upper bounding scheme (1990) is used with slight modifications to compute a bound on the number of visited vertices, which permits to update the integer linear programming formulation at each node, tightening the linear programming bound. The procedure is successfully tested on instances involving between 10 and 15 vertices.

5.3 The assignment related upper bound and algorithms

A classical relaxation for the TSP consists in relaxing subtour elimination constraints (4). The resulting assignment problem can then easily be solved. In the STSP context, relaxing subtour elimination constraints induces a problem still difficult to solve because of knapsack constraint (5). However, Kataoka and Morito (1988) propose a Branch and Bound procedure facing this difficult problem for deriving upper bounds. This procedure is detailed in the following subsection. In an other way, a weaker upper bound can still easily be found with the help of linear programming by relaxing integrality constraints (7) and (8). Although weaker, such a bound can be strengthened with valid inequalities. Attempts in this direction are described further.

5.3.1 Kataoka and Morito's Branch and Bound procedure

The Branch and Bound scheme proposed by Kataoka and Morito (1988) derives from classical Branch and Bound procedures for TSPs. Indeed, authors mainly focus on an efficient way to compute upper bounds. As indicated previously, they relax subtour elimination constraints, but the relaxed problem is still difficult to solve. They propose to solve it with a Branch and Bound procedure managing branches efficiently by taking account of the special structure of the cost function. In this Branch and Bound procedure, the upper bounding scheme consists in relaxing integrality constraints (7) and (8). To be more efficient, authors solve it as a minimum cost flow problem with an additional constraint and use a specific method developed by Kobayashi (1983).

Computational results are limited to asymmetric instances involving 10 vertices, which are solved in less than two minutes on a PC NEC-9801VM2 (10Mhz). Results show that instances are more difficult to solve when the time constraint implies the visit of about half of the vertices.

5.3.2 Procedures introducing valid inequalities

Leifer and Rosenwein (1994) propose to tighten this bound with valid inequalities. Fischetti *et al.* (1998) and Gendreau *et al.* (1998a) develop several classes of valid inequalities and use them in a Branch and Cut algorithm. These valid inequalities are especially extensions of known valid inequalities for the TSP. Knapsack constraint (27) also permits the use of valid knapsack problem inequalities. Fischetti *et al.* (1998) sort these inequalities as logical constraints, matching inequalities, cover inequalities, path inequalities and conditional cuts (used in the cutting-plane context).

Leifer and Rosenwein's (1994) upper bound is evaluated on Tsiligirides' instances (1984) (involving up to 33 vertices) and performs with accuracy in no more than a few seconds on a *KORBX* Processor, but this bound is not embedded in a search tree.

Branch and Cut algorithms prove to be very efficient. Instances involving up to 300 vertices are solved to optimality by Gendreau *et al.* (1998a) in less than 3 hours on a Sun Sparcstation 1000, while Fischetti *et al.* (1998) succeed in solving instances involving up to 500 vertices in a few more hours on an HP Apollo 9000/720 computer.

5.4 1-Tree relaxation style bounds and related algorithms

The formulation introduced by Ramesh *et al.* (1992) (see section 5.1) is handy to adapt the 1-Tree relaxation previously used for the TSP. Indeed, relaxing in a Lagrangian fashion constraints related to the time limit and degree requirements on the vertices lead to a 1-Tree formulation (tree rooted at the depot plus one additional arc back to the depot). This relaxation is used within a Branch and Bound procedure to obtain an exact solution for the undirected STSP. At each node of the search tree, a bound is computed by a subgradient optimization improving the Lagrangian multipliers. The authors focus on some points providing the speeding up of the solution procedure. Lagrangian multipliers are initially selected with dual considerations. A scaling mechanism is developed to avoid an excessive emphasis on one of the two types of violations involved by the relaxation. The lower bound is improved with a heuristic procedure. Parameters are chosen with a fine tuning. This Branch and Bound procedure behaves well for instances having up to 150 vertices and gives optimal solutions in less than 5 minutes on an IBM3081/GX.

In the same way, the minimum directed 1-subtree upper bound is used by Kataoka *et al.* (1998) in the directed case. They begin to propose a Lagrangian relaxation for the STSP. Constraints (2) and (5) are relaxed and included into the objective function with Lagrangian multipliers. They are able to add constraints $x_{ij} \leq y_i$ for each arc $(v_i, v_j) \in A$, as these constraints are valid for the STSP. The Lagrangian relaxation problem then consists in finding a directed 1-subtree (directed subtree rooted at the depot plus an additional arc back to the depot) of minimal cost. As the minimum directed 1-subtree problem (MD1SP) is NP-Hard, they focus on finding strong lower bounds for this problem. With this goal, they first add cuts and perform a dual simplex procedure. They then apply Lagrangian relaxation. This strategy proved to be efficient for finding strong lower bounds for MD1SP. For the STSP, these bounds seem to outperform those obtained from the ordinary assignment relaxation. The Lagrangian framework built up to solve the STSP is left for future investigations. Table 3 summarizes exact solution procedures developed for solving the STSP and sizes of the larger instances solved.

Reference	Technique	Number of vertices	
		Symmetric	Asymmetric
Kataoka and Morito (1988)	Branch and Bound		10
Laporte and Martello (1990)	Branch and Bound	70	90
Ramesh *et al.* (1992)	Branch and Bound	150	150
Millar and Kiragu (1997)	Branch and Bound	15	15
Gendreau *et al.* (1998a)	Branch and Cut	300	
Fischetti *et al.* (1998)	Branch and Cut	500	

Table 3: Exact solution procedures for the STSP.

6 Extensions

Up to this section, we have focused on solution strategies for the pure STSP. However, many papers deal with extensions of the STSP, altered with some new features. The most

significant modification consists in introducing a fleet of vehicles for collecting the profit instead of a single vehicle. We then speak of multi-vehicle selective routing problems, which we detail in section 7. In this section, we focus on single-vehicle extensions of the STSP.

6.1 Time Relaxed STSP

In the Time relaxed STSP, a penalty is subtracted from the profit when travel time exceeds the time limit. Golden *et al.* (1986) propose two heuristic procedures:

1. **Incremental algorithm.** A STSP solution procedure is used for a number of values of t_{max} exceeding the time limit. Costs are updated according to the penalties and the best solution is selected.

2. **Insert/delete algorithm.** A STSP solution procedure is used with a time limit somewhat less than the true t_{max}. New vertices are inserted depending on their profitability (profit minus penalty occurred). Vertices are deleted when the route exceeds the time limit and their deletion would improve the solution.

The incremental algorithm proves to be clearly superior.

6.2 STSP with Time Windows

In the STSP with Time Windows, the visit of vertices is constrained by time windows. Kantor and Rosenwein (1992) propose two approaches.

The first approach consists in adapting insertion procedures by taking into account the time window constraints. A vertex is inserted only when the solution remains feasible. The second approach is called tree heuristic. An enumeration tree is built up. To accelerate the procedure, some rules are introduced. The maximal travel time between two vertices and the maximum allowable waiting time are limited. Partial paths are abandoned when they are not sufficiently promising.

These two heuristic procedures are evaluated on Tsiligirides' instances (1984) involving up to 33 vertices, which have been adapted to include time windows. The second approach appears more effective as long as time windows or t_{max} strongly constrain the instances. Let us point out that the STSP with Time Windows is also studied by Cavalier and Tranguch (1995), who propose another heuristic solution procedure.

6.3 STSP with Time-dependent Rewards

In the STSP with Time-dependent rewards, profits decrease as time passes. This extension of the STSP is introduced by Brideau and Cavalier (1994) and by Erkut and Zhang (1996). This might be seen as an adaptation of the STSP in a competitive environment. For instance, profits might represent sales and decline over time since several salespeople are operating in the same area. It might also be relevant when profits correspond to services with a value depending on delays (e.g., for an equipment repair company).

Brideau and Cavalier (1994) propose a greedy construction heuristic, improved with a 2-opt procedure. Roughly speaking, vertices are added depending on a profit-per-time unit measure (actually measure involves pairs of vertices in a somewhat complicated way).

Erkut and Zhang (1996) propose another route extension criterion. A penalty is associated to every non-visited vertex, related to a possible decrease of its profit. To this end, the procedure measures the minimum loss of the collected profit for a vertex if it is collected after any other one rather than immediately. This measure is divided by the travel time consumed for adding the vertex to the route and then provides the value of the penalty. Thus, a vertex with a high penalty should be added as soon as possible and is selected for the next route extension. This greedy procedure proves to generally outperform the previous one in a faster computing time. One must note though that, in both cases, greedy extensions are fast and that most of the time is consumed during the 2-opt improvement phase concluding the procedures.

In order to evaluate the accuracy of these heuristics, Erkut and Zhang (1996) develop an exact Branch and Bound solution procedure based on the enumeration of all possible routes. A simple upper bounding scheme based on the remaining profit is used for pruning nodes. This exact procedure manages to solve instances with up to 20 vertices in less than 2 hours on a 60-Mhz Pentium microcomputer. It indicates that the average optimality of the solutions produced by the penalty heuristic is about 96% (the worst solution being about 90%). Using sequentially both heuristics permits to improve these results respectively to 98% and 96%.

6.4 STSP in non-Euclidean graphs

As indicated at the beginning of this survey, we consider that the pure STSP is defined on a complete graph with a positive distance matrix satisfying the triangle inequality. However, some papers deal with STSPs defined on data out of this framework. In particular, Ramesh and Brown (1991) devise their four-phase heuristic procedure (see section 4.4.2) for any general distance matrix. Moreover, they consider variants of the STSP allowing or not revisitation of vertices (but with profits collected only once). Even if such considerations do not seem to have a direct practical utility, non-Euclidean STSPs might always occur, for example, as subproblems.

6.5 Other variants

Several other extensions of the STSP are proposed by different authors:

1. Gendreau *et al.* (1998a) generalize the STSP by introducing a compulsory set of vertices. It defines a compromise between STSP and TSP.

2. Awerbuch *et al.* (1998) propose to suppress the depot condition. It enables them to adapt an approximation algorithm initially dedicated to the Quota TSP.

3. Golden *et al.* (1997) and Wang *et al.* (1996) generalize the STSP to a situation where profits are evaluated with respect to several attributes (natural beauty, historical significance, cultural and educational attractions and business opportunities) and the objective function is nonlinear. A neural network solution procedure is developed and applied to a problem involving 27 cities in China.

4. Deitch and Ladany (2000) are interested in the determination of an attractive bus route in a tourist region. With this goal, they generalize the STSP by introducing

profits on arcs (denoting scenic route segments) and service times on vertices. Additionally, repeated visits are possible but profits and service times are only taken into account the first time. The authors describe a transformation into the classical STSP and propose a solution algorithm based on insertions and deletions. This algorithm is rather similar to the one by Mittenthal and Noon (1992), that is designed for the solution of the TSSP+1.

Besides the previous variants, one of the engineering design problems at NASA has been formulated as a version of the STSP, and has been posed as a design challenge problem by the American Institute of Aeronautics and Astronautics (AIAA). It consists in a STSP with several side constraints (e.g., precedence constraints) and stochastic components. Many attempts were devised for solving the challenge. DeJong (1988) and Press and Callan (1988) propose heuristic solution procedures essentially driven by insertion, deletion or Monte Carlo method. Though not referring to earlier attempts for solving the STSP (Tsiligirides (1984), Golden *et al.* (1987)), these procedures are based on similar principles. Apart these, some tree search solution approaches can also be found. Lirov (1988) guides the search with an expert system, while Shaw *et al.* (1988) use bounding schemes and heuristic components in somewhat of a Branch and Bound fashion. Unfortunately, the challenge problem introduces too many features to really offer new insights into the real STSP.

7 Multi-vehicle selective routing problems

As in the TSP context, many realistic situations involve several vehicles. Reproducing TSP terminology, a pure m-STSP would consist in determining m simple circuits including vertex v_1 of maximal total profit, whose durations do not exceed a preset value t_{max} and such that no two vehicles visit a same vertex (excepting the depot). In fact, most situations involve any type of constraints and give rise to a whole family of problems which we could compare to the VRP family. However, we have to point out that the selection issue and the objective function induce some specificities. In particular, since the travel time does not penalize the objective function, an optimal policy consists in using every available vehicle. Thus, unlike the VRP, the restriction to m vehicles always plays a prominent part.

It is in the field of military applications that the first attempts to deal with multi-vehicle selective routing problems appear. Moser (1990) proposes a heuristic and an exact solution procedure for a tactical aerial reconnaissance problem. This problem takes time windows into account and is called the Multi-Player Orienteering Problem with Time-Windows. Wu (1992) and Dunn (1992) face another military application. Logistics ships are used on a dispersed battle group to travel long distances between ships and replenish them. When operational requirements limit the amount of available time, replenished ships are selected depending on a combat value added to the group by replenishment. Dunn (1992) proposes a dynamic programming approach running quickly enough to be useful for scheduling underway replenishment in operational situations.

As it happens in the TSP context, some STSP heuristics can effectively be extended to the multi-vehicle situation. Chao *et al.* (1996b) propose a straightforward extension of their heuristic developed for the OP (see section 4.4.3). They call the problem the Team Orienteering Problem, as it still concerns orienteering. A team tries to collect as much

profit as possible instead of a single competitor in usual score orienteering events. Once a team member has collected the award associated to a location, no other team member can be awarded a profit for visiting the same location. Millar and Kiragu (1997) extend their formulation and their Branch and Bound solution procedure (see 5.1 and 5.2) for the multi-vehicle case. They consider both time limits for each vehicle and for the total time over all vehicle tours. The context is a fisheries patrol problem. A deterrent value (depending on fishing factors and historical patrol activity for the zone) is associated with each fishing zone and patrol routes are computed so as to maximize the deterrent value of the patrols while respecting time limits.

Besides the previous papers, one can find some other attempts for solving multi-vehicle selective routing problems. Butt and Cavalier (1994) are concerned with recruiting athletes. A college recruiter has to recruit football players from area high schools in a given number of days. Each day, he has a limited amount of time to visit some chosen high schools and come back home. A reward is assigned to each high school based on its recruiting potential. The objective is to visit a set of high schools maximizing the recruiting potential. They propose a greedy algorithm for solving this problem. Butt and Ryan (1999) face the same problem and propose a column generation solution procedure. Gueguen (1999) also proposes a column generation solution procedure for the so called Selective Vehicle Routing Problem with Time Windows (Selective VRPTW), where vertices stand for customers and where vehicle capacity, service times and time windows for customers are introduced. These three last procedures are described below.

At last, we have to mention that the special case of 2 vehicles is studied by Nocito (1993) who proposes heuristic solution procedures. Table 4 summarizes attempts for solving multi-vehicle selective routing problems.

Reference	Context	Solution approach
Moser (1990)	aerial reconnaissance problem	heuristic + exact solution procedure
Wu (1992)	underway replenishment	
Dunn (1992)	underway replenishment	dynamic programming
Nocito (1993)		heuristics
Butt and Cavalier (1994)	recruiting athletes	greedy heuristic
Chao *et al.* (1996b)	team orienteering	heuristic
Millar and Kiragu(1997)	fisheries patrol	Branch and Bound
Butt and Ryan (1999)	recruiting athletes	column generation
Gueguen (1999)		column generation

Table 4: Multi-vehicle selective routing problems.

7.1 A greedy heuristic procedure

Butt and Cavalier (1994) propose a measure W_{ij} for evaluating the interest of visiting two vertices v_i and v_j in a same route:

$$W_{ij} = \alpha \frac{p_i + p_j}{t_{max}} t(v_1, v_i, v_j, v_1) + (1 - \alpha) \frac{p_i + p_j}{t(v_1, v_i, v_j, v_1)} t_{max}$$

where $0 \leq \alpha \leq 1$ and $t(v_1, v_i, v_j, v_1)$ is the duration of a route visiting only vertices v_i and v_j. This measure is defined to emphasize either highly profitable close vertices close to the depot or distant highly profitable vertices, depending on the value of α. This measure is used in a greedy way to build up the routes. The routes are built up one by one by inserting vertices belonging to a vertex pair with a maximal measure and such that exactly one of its vertices is visited, performing the insertion with a TSP insertion heuristic and checking for feasibility. A complementary step checks that it would not be more profitable to visit a single very profitable vertex instead of the freshly built up route. This step is necessary as the procedure always begins new routes with two vertices (the ones from the vertex pair with the higher measure).

The complete heuristic consists of a succession of solutions with different α values. At the end, the best solution is selected. This procedure performs very effectively on small instances. Computational results are given for instances involving up to 100 vertices, but no comparison with exact solutions is given for large instances. Results show that the α value has a great impact on the solution value. For small instances, the best solution happens for any α value, depending on the distribution of the vertices. But as instances grow, small α values outperform the others. In our opinion, this behaviour indicates that measures with $\alpha = 0$ reflect better the interest of a pair of vertices, but, as shown by small instances, does not catch all the interesting vertices. Thus, we presume that the heuristic is not able to solve large instances as well as small instances.

7.2 Column generation procedures

Butt and Ryan (1999) and Gueguen (1999) adapt the classical set partitioning formulation used in column generation solution procedures for the VRP to two multi-vehicle selective routing problems. They propose the same following model:

$$\text{maximize} \sum_{k \in \Omega} c_k x_k \tag{32}$$

subject to

$$\sum_{k \in \Omega} a_{ik} x_k \leq 1 \qquad (i \in \Omega), \tag{33}$$

$$\sum_{k \in \Omega} x_k \leq m, \tag{34}$$

$$x_k \in \{0, 1\} \qquad (k \in \Omega), \tag{35}$$

where Ω is the set of feasible routes, index k represents the routes, c_k is the profit of route k and a_{ik} is equal to 1 if route k visits customer i and 0 otherwise. $x_r = 1$ if and only if route r is part of the solution. Objective function (32) is to maximize the collected profit on the tours taken through the graph. Constraints (33) ensure that each customer is visited at most once and constraint (34) limits the number of vehicles used.

Let α_i be the dual variable associated to the packing constraint (33) for vertex v_i ($\alpha_i \geq 0$), and Γ ($\Gamma \geq 0$) be the dual variable associated to constraint (34). The subproblem consists in finding a feasible route of profit strictly better than Γ, where profits on vertices

are changed to $p_i - \alpha_i$. Butt and Ryan's approach for solving this subproblem is different than Gueguen's, as detailed below.

Note that both feasible route sets have their distinctions. For Butt and Ryan (1999), the fleet of vehicles is heterogeneous (the time limit depends on the type of vehicle) and a service time for the vertices is taken into account. For Gueguen (1999), a vehicle capacity constraint, time windows and service times for the clients are introduced.

7.2.1 Butt and Ryan's subproblem approach

Butt and Ryan (1999) begin to rank vertices according to their (new) profits. Vertices with negatives profits are put aside as they cannot improve the profit of a route. The principle of the algorithm is to scan the set of vertex subsets containing the depot and to solve a TSP on a subset when its profit is high enough. The set of vertex subsets is scanned in a lexicographic order (the depot is implicitly considered). When a TSP solution goes over the time limit, the child subsets are not considered. Subsets S such that $|S| > 2$ are also checked for feasibility to avoid assessing many long infeasible routes when profits are low. The enumeration procedure stops either when it is completed or $10n$ good routes have been found or all children of a subset $\{v_i\}$ have been scanned and at least one good route has been found. TSPs are solved with an existing exact procedure. An important matter is that each time the feasibility question is solved for a subset of vertices, the result is kept and used at further iterations.

7.2.2 Gueguen's subproblem approach

Gueguen (1999) tackles the subproblem as an Elementary Shortest Path Problem with Resource Constraints. He claims that, unlike subproblem solution for the classical VRP with Time Windows, the elementary restriction is necessary here, as visiting many times a customer would give rise to repeated profits. A dynamic programming solution procedure proposed by Gueguen *et al.* (2000) is used. It extends dynamic programming procedures dedicated to the non elementary Shortest Path Problem with Resource Constraints, with an expanded number of states but a dominance criterion enforced to reduce it.

7.2.3 Branch and Price

In both approaches, the column generation scheme is embedded into a Branch and Price procedure. Gueguen (1999) branches on arcs. An arc (v_i, v_j) with a fractional value in the relaxed solution is either imposed or forbidden. Butt and Ryan (1999) branch on a vertex pair (v_i,v_j). Either both vertices are imposed, or at most one of them is accepted. The pair is selected among pairs that maximize $\sum_{k\in\Omega} a_{ik}a_{jk}x_k$ with $\sum_{k\in\Omega} a_{ik}a_{jk}x_k < 1$. Ties are broken by choosing the pair with a largest score-to-saving ratio.

Butt and Ryan (1999) manage to solve instances involving up to 100 customers and 15 vehicles in a few hours. With the time windows minimizing the set of feasible routes, Gueguen (1999) manages to solve instances involving up to 100 customers and 10 vehicles in less than a hour.

8 Conclusion

In this paper, we have proposed a classification and a commented survey regarding the Selective Traveling Salesman Problem (STSP) as well as some extensions. The STSP is a generalization of the Traveling Salesman Problem (TSP) where it is not necessary to visit all vertices. With each vertex is associated a profit. The objective is to find a route which maximizes the collected profit in a time not exceeding a given bound. Golden *et al.* (1981) claim that this situation is often a more realistic statement of the problem a traveling salesman faces than the basic TSP. Moreover it occurs in a wide variety of other situations like military, environmental or industrial ones. Many solution procedures have been proposed, extending the major fields of investigation for solving the TSP. In particular, classical construction, improvement or linear relaxation based heuristics have been addressed. The problem structure and its special features (especially the possibility of dropping a vertex) have also favored the development of metaheuristic approaches. The multi-vehicle version has seen less consideration, but lately efficient procedures appeared.

We believe that more and more situations will be considered as STSP related problems in the future and efforts have to be done in the present direction, especially in the field of multi-vehicle selective routing problems.

References

Awerbuch B. , Azar Y., Blum A., and Vempala S. New approximation guarantees for minimum-weight k-trees and prize-collecting salesmen. *SIAM Journal on Computing*, 28(1):254–262, 1998.

Balas E. The Prize Collecting Traveling Salesman Problem. II: Polyhedral results. *Networks*, 25(4):199–216, 1995.

Brideau R.J. III and Cavalier T.M. The Maximum Collection Problem with Time Dependent Rewards. Presented at TIMS International conference, Alaska, June 1994.

Butt S.E. and Cavalier T.M. A heuristic for the Multiple Tour Maximum Collection Problem. *Computers & Operations Research*, 21(1):101–111, 1994.

Butt S.E. and Ryan D.M. An optimal solution procedure for the Multiple Tour Maximum Collection Problem using column generation. *Computers & Operations Research*, 26(4):427–441, 1999.

Cavalier T.M. and Tranguch B.A. The Maximum Collection Problem with Time Windows. Presented at INFORMS New Orleans Conference, November 1995.

Chao I.-M., Golden B.L., and Wasil E.A. A fast and effective heuristic for the Orienteering Problem. *European Journal of Operational Research*, 88(3):475–489, 1996a.

Chao I.-M., Golden B.L., and Wasil E.A. The Team Orienteering Problem. *European Journal of Operational Research*, 88(3):464–474, 1996b.

Cloonan J. A heuristic approach to some sales territory problems. In Little JDC, editor, *Proceedings of the Fourth International conference on Operations Research*, pages 81–84, Cambridge, Massachusetts, USA, 1966. MIT Press.

Dantzig G.B., Fulkerson D.R., and Johnson S.M. Solution of a large-scale Traveling-salesman Problem. *Operations Research*, 2:393–410, 1954.

Davis S., Ceto N. Jr, and Rabb J.M. A comprehensive check processing simulation model. *Journal of Bank Research*, 13:185–194, 1982.

Deitch R. and S.P. Ladany. The One-period Bus Routing Problem: solved by an effective heuristic for the Orienteering Tour Problem and improvement algorithm. *European Journal of Operational Research*, 127:69–77, 2000.

DeJong C.D. Rapid solution of constrained traveling salesman problems. *Journal of Guidance, Control, and Dynamics*, 11(5):403–405, 1988.

Dell'Amico M., Maffioli F., and Värbrand P. On prize-collecting tours and the Asymmetric Travelling Salesman Problem. *International Transactions in Operational Research*, 2(3):297–308, 1995.

Dueck G. New optimization heuristics: The great deluge algorithm and the record-to-record travel. *Journal of computational physics*, 104(1):86–92, 1993.

Dunn J.S. Scheduling underway replenishment as a generalized orienteering problem. Master's thesis, Naval Postgraduate School, Monterey, California, USA, 1992.

Erkut E. and Zhang J. The Maximum Collection Problem with Time-dependent Rewards. *Naval Research Logistics*, 43(5):749–763, 1996.

Feillet D., Dejax P., and Gendreau M. Traveling salesman problems with profits: An overview. Technical Report CER 00-04A, Laboratoire Productique Logistique, Ecole Centrale Paris, 2000.

Fischetti M., Salazar González J.J., and Toth P. Solving the Orienteering Problem through Branch-and-Cut. *INFORMS Journal on Computing*, 10(2):133–148, 1998.

Fischetti M. and Toth P. An additive approach for the optimal solution of the Prize-collecting Travelling Salesman Problem. In Golden B.L. and Assad A.A., editors, *Vehicle Routing: Methods and Studies*, pages 319–343. Elsevier Science Publishers B.V. (North-Holland), 1988.

Gendreau M., Hertz A., and Laporte G. New insertion and postoptimization procedures for the Traveling Salesman Problem. *Operations Research*, 40:1086–1094, 1992.

Gendreau M., Laporte G., and Semet F. A Branch-and-Cut algorithm for the Undirected Selective Traveling Salesman Problem. *Networks*, 32:263–273, 1998a.

Gendreau M., Laporte G., and Semet F. A tabu search heuristic for the Undirected Selective Traveling Salesman Problem. *European Journal of Operational Research*, 106:539–545, 1998b.

Gensch D.H. An industrial application of the Traveling Salesman Subtour Problem. *AIIE Transactions*, 10(4):362–370, 1978.

Golden B.L., Assad A., and Dahl R. Analysis of a large scale vehicle routing problem with an inventory component. *Large Scale Systems*, 7:181–190, 1984.

Golden B.L., Levy L., and Dahl R. Two generalizations of the Traveling Salesman Problem. *OMEGA*, 9(4):439–441, 1981.

Golden B.L., Levy L., and Vohra R. The Orienteering Problem. *Naval Research Logistics*, 34(3):307–318, 1987.

Golden B.L., Storchi G., and Levy L. A time-relaxed version of the Orienteering Problem. Technical Report Series MS/S 86-014, College of Business and Management, University of Maryland, 1986.

Golden B.L., Wang Q., and Liu L. A multifaceted heuristic for the Orienteering Problem. *Naval Research Logistics*, 35:359–366, 1988.

Golden B.L., Wang Q., and Sun X. A neural network solution to the Generalized Orienteering Problem. Presented at INFORMS Dallas Conference, October 1997.

Göthe-Lundgren M., Jörnsten K., and Värbrand P. On the nucleolus of the basic vehicle routing game. *Mathematical Programming*, 72:83–100, 1996.

Gueguen C. *Méthodes de résolution exacte pour les problèmes de tournées de véhicules.* PhD thesis, Laboratoire Productique Logistique, Ecole Centrale Paris, 1999.

Gueguen C., Dejax P., Dror M., Feillet D., and Gendreau M. An exact algorithm for the Elementary Shortest Path Problem with Resource Constraints: Application to some vehicle routing problems. Technical Report CER 00-02a, Laboratoire Productique Logistique, Ecole Centrale Paris, 2000.

Kantor M.G. and Rosenwein M.B. The Orienteering Problem with Time Windows. *Journal of the Operational Research Society*, 43(6):629–635, 1992.

Kataoka S. and Morito S. An algorithm for the Single Constraint Maximum Collection Problem. *Journal of the Operations Research Society of Japan*, 31(4):515–530, 1988.

Kataoka S., Yamada T., and Morito S. Minimum directed 1-subtree relaxation for Score Orienteering Problem. *European Journal of Operational Research*, 104:139–153, 1998.

Keller C.P. Algorithms to solve the Orienteering Problem: A comparison. *European Journal of Operational Research*, 41:224–231, 1989.

Kobayashi T. The lexico-shortest route algorithm for solving the minimum cost problem with an additional linear constraint. *Journal of the Operations Research Society of Japan*, 26(2):167–185, 1983.

Laporte G. and Martello S. The Selective Travelling Salesman Problem. *Discrete Applied Mathematics*, 26:193–207, 1990.

Leifer A.C. and Rosenwein M.B. Strong linear programming relaxations for the Orienteering Problem. *European Journal of Operational Research*, 73:517–523, 1994.

Lin S. Computer solutions of the Traveling Salesman Problem. *Bell System Computer Journal*, 44:2245–2269, 1965.

Lirov Y. Expert systems approach for generalized traveling salesman problem. *Journal of Guidance, Control, and Dynamics*, 11(5):425–429, 1988.

Millar H.H. and Kiragu M. A time-based formulation and upper bounding scheme for the Selective Travelling Salesperson Problem. *Journal of the Operational Research Society*, 48(5):511–518, 1997.

Mittenthal J. and Noon C.E. Insert/delete heuristic for the Travelling Salesman Subsettour Problem with One Additional Constraint. *Journal of the Operational Research Society*, 43(3):277–283, 1992.

Moser H.D. Jr. Scheduling and routing tactical aerial reconnaissance vehicles. Master's thesis, Naval Postgraduate School, Monterey, California, USA, 1990.

Nocito J.B. Heuristic solutions to the Two-player Orienteering Problem. Master's thesis, University of Maryland at College Park, 1993.

Pekny J.F. and Miller D.L. An exact parallel algorithm for the Resource Constrained Travelling Salesman Problem with application to scheduling with an aggregate deadline. In *ACM Eighteenth Annual computer Science Conference*, 1990.

Pekny J.F., Miller D.L., and McRae G.J. An exact parallel algorithm for scheduling when production costs depend on consecutive system states. *Computers and Chemical Engineering*, 14:1009–1023, 1990.

Press W.H. and Callan C.G. Jr. Efficiency-guided method for traveling-salesman-like resource allocation problems. *Journal of Guidance, Control, and Dynamics*, 11(5):394–396, 1988.

Ramesh R. and Brown K.M. An efficient four-phase heuristic for the Generalized Orienteering Problem. *Computers & Operations Research*, 18(2):151–165, 1991.

Ramesh R., Yoon Y-S., and Karwan M.H. An optimal algorithm for the Orienteering Tour Problem. *ORSA Journal on Computing*, 4(2):155–165, 1992.

Rosenkrantz D.J., Stearns R.E., and Leis P.M. An analysis of several heuristics for the Traveling Salesman Problem. *SIAM Journal on Computing*, 6:563–581, 1977.

Shaw J.J., James R.M., and Grunberg D.B. Birth of a salesman. *Journal of Guidance, Control, and Dynamics*, 11(5):415–420, 1988.

Smith A.E. and Tasgetiren F.M. A genetic algorithm for the Orienteering Problem. Presented at INFORMS Philadelphia Conference, November 1999.

Tsiligirides T. Heuristic methods applied to orienteering. *Journal of the Operational Research Society*, 35(9):797–809, 1984.

Wang Q., Sun X., and Golden B.L. Using artificial neural network to solve generalized orienteering problems. In *Proceedings of the 1996 Artificial Neural Networks in Engineering*, 1996.

Wang Q., Sun X., Golden B.L., and Jia J. Using artificial neural networks to solve the Orienteering Problem. *Annals of Operations Research*, 61:111–120, 1995.

Wren A. and Holliday A. Computer scheduling of vehicles from one or more depots to a number of delivery points. *Operations Research Quarterly*, 23:333–344, 1972.

Wu T. Optimization models for underway replenishment of a dispersed carrier battlegroup. Master's thesis, Naval Postgraduate School, Monterey, California, USA, 1992.